An Energy-Efficient Method with Dynamic GPS Sampling Rate for Transport Mode Detection and Trip Reconstruction

Presented by Jonathan Milot

Authors : Jonathan Milot, Jaël Champagne Gareau and Eric Beaudry

UQÀM

caiac
CANADIAN ARTIFICIAL INTELLIGENCE ASSOCIATION
ASSOCIATION POUR L'INTELLIGENCE ARTIFICIELLE AU CANADA

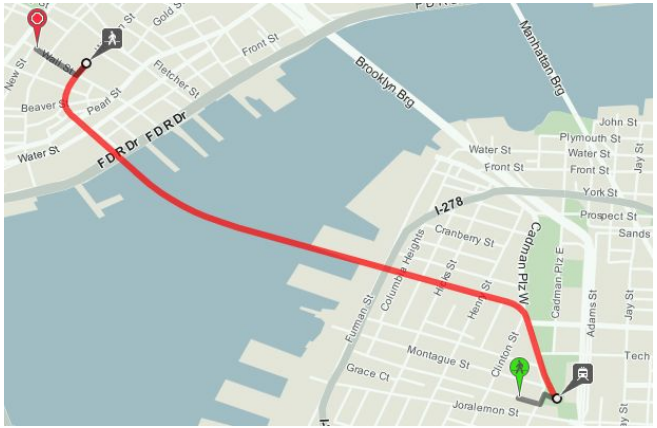
Fonds de recherche
Nature et
technologies
Québec

Discussed points

- Problem definition
- Approaches and algorithms typically used for :
  - Trip Reconstruction
  - Transport Mode Detection
  - Combined Approach
  - Energy consumption
- Model and Algorithm
- Experimentations setup
- Results
  - Accuracy
  - Energy Consumption
- Conclusion
- References

Problem definition

- Trip reconstruction (TR)
  - What is the sequence of nodes crossed to get from Point A to Point B ?
- Transport Mode Detection (TMD)
  - What mode (walk, bus, car, etc.) was used to transition on each part of the trip ?



OpenTripPlanner, a multi modal trip planner (http://www.opentripplanner.org/).

- This is the inverse problem of trip planning
  - Plan recognition versus Planning

Trip Reconstruction

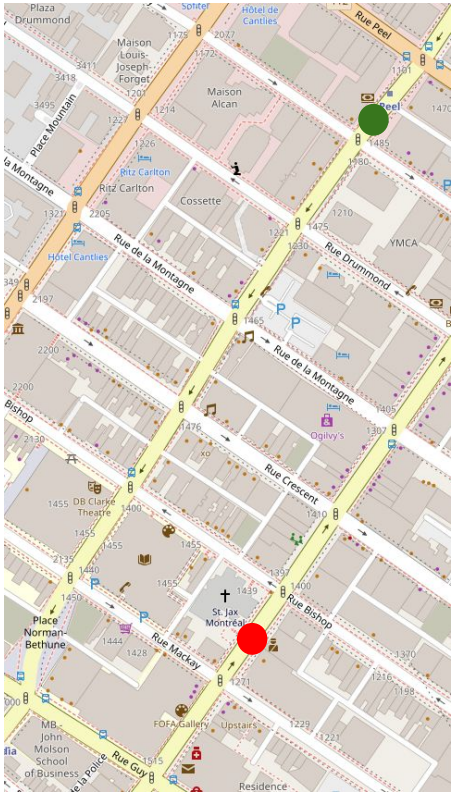
The current approaches use a fixed GPS sampling rate.

- Every 1s, 10s, 60s ? Find the one minimizing the error.
- [11] simply tries to match each GPS reading to the nearest OSM node
  - Doesn’t work ? Try with the second nearest, etc.
- [10] uses a Hidden Markov Model (HMM) to determine the most likely road segment for each GPS reading.

Trip Reconstruction

GPS sampling rates are kept low (< 60s)

- Avoid *arc-skipping* problems as much as possible
- Multiple paths explain the transition between two GPS lectures



Which paths explain the best the transition between the two GPS reading ?

Transport Mode Detection

Machine Learning is generally used to detect the transport mode, based on GPS data (position, speed).

The goal : Classify data (GPS) amongst a defined and limited set of classes (transport modes).

Algorithms used : Neural Network [3,15], Convolutional Neural Network [9], Random Forest [13], Support Vector Machine [2]

They generally achieve a good accuracy (> 90%), but rely on a high GPS sampling rate (from 1 to 15 seconds interval) and do not attempt to reconstruct the smartphone’s trip.

Transport Mode Detection - Accuracy

Table 1. Transport mode detection accuracy in related works

Approach	Sampling rate (s)	Walk	Car	Bus	Average
[13]	15	98.9%	80.8%	93.0%	93.8%
[2]	60	93.8%	88.5%	58.3%	88.0%
[3]	1	95.0%	72.0%	84.0%	83.8%
[15]	1	98.5%	94.2%	88.4%	94.4%
[9]	1–5	95.7%	67.4%	81.1%	84.8%

Combined Approach

TR and TMD at the same time :

- [8] uses a conventional GIS-based map-matching algorithm to reconstruct the path and a rule-based algorithm to identify transport modes (walk, bicycle, bus and car).
- TR error : 21%
- TMD accuracy : 92 % on average

Energy consumption

GPS sensor is energy expensive :

- HTC Dream and Google Nexus One : 1 Hz consumes 143.1 mW [5]
- In our experiments :
  - Samsung Galaxy S8 : 439.3 mW
  - Asus Zenphone 4 Max : 397.44 mW

Battery only lasted 8 to 10 h for normal daily usage with GPS on.

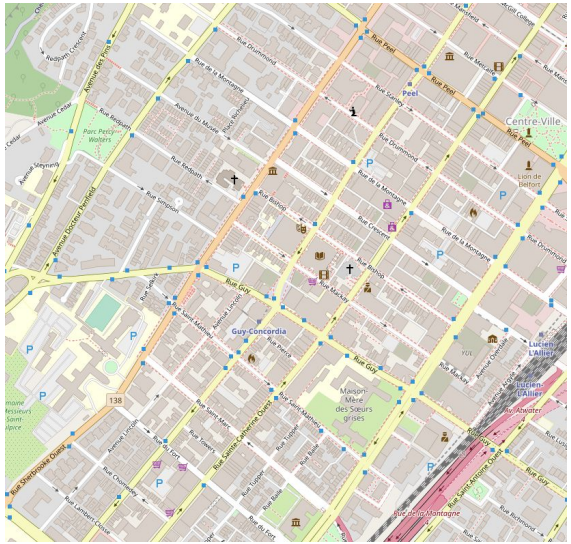
Model and Algorithm

Objective :

- TR and TMD at least as good as Related Works
- Reduce the energy consumption
    - It needs to be acceptable for mass usage by multiple users

How? Dynamic sampling rate.

Model and Algorithm



Dense urban area
Lot of transport modes,
lot of roads

Suburban area
Less transport modes,
Still lot of roads

Rural area
Few transport modes,
Few roads

Model and Algorithm

The actual optimal sampling rate depends on many factors :

- Underlying road network configuration
- Accessible transport modes nearby
    - Schedules, stops emplacements, etc.

Having a fixed sampling rate means :

- Wasting energy if the new sensing won’t gives new information
- Missing important informations

Model and Algorithm

We use a particle filter to estimate the smartphone's state according to GPS readings made at a dynamic rate.

Overview of our method :

1. Make a GPS reading to estimate the smartphone's state s.
2. Simulate the evolution of s until a time t in the future.
3. Determine a time $t^* \in]0, t]$ offering an interesting compromise between accuracy and energy consumption. When t^* is reached, return to 1.

Model and Algorithm

Formally :

- The world is a graph : G = (N,A).
- n = (n.Lat, n.Long) $\in$ N, a = (n_{from}, n_{to}, v_{max}, T_e) $\in$ A
- We search the sequence of nodes and transport modes P, where
  - P = <(n_0, m_0),(n_1, m_1),...,(n_k-1, m_k-1),(n_k, m_k)>, where m_i is the transport mode used to reach n_i
- Smartphone's state *s* at time *t* : s_t = (*p, m, v, P*)
  - *p* = postion
  - *m* = current transport mode
  - *v* = velocity
  - *P* = path travelled

Model and Algorithm

However, the smartphone's state *s* is not directly observable, but is estimated from an external sensor (GPS).

- The state position $s.p \sim N(x, \sigma^2)$
  - x is the GPS coordinate projection on the nearest arc
  - σ^2 is the accuracy of the GPS.
- The evolution of $s.p$ over a time interval Δt is :
  - $s_{t+\Delta t.p} = s_t.p + N(s.v, \sigma^2) \times \Delta t$
  - Indeed, $s.v$ can vary during Δt (traffic, lights, etc.)

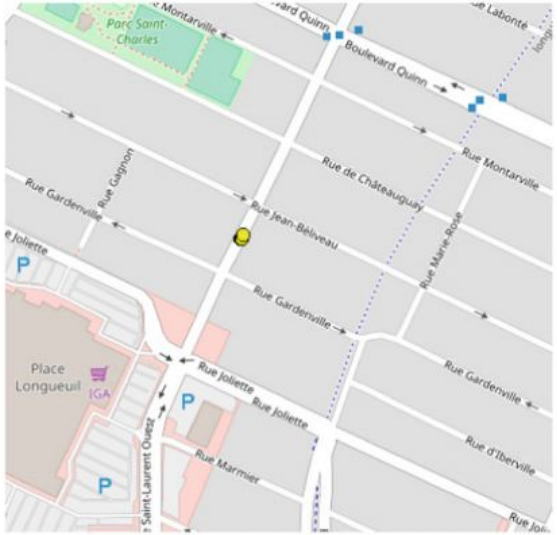
Model and Algorithm

- Approximate a belief state $bel(x_t)$ by a set of particles X_t constructed by the control data u_t and the sensor data z_t
- u_t = Δt since last GPS
- z_t = GPS reading

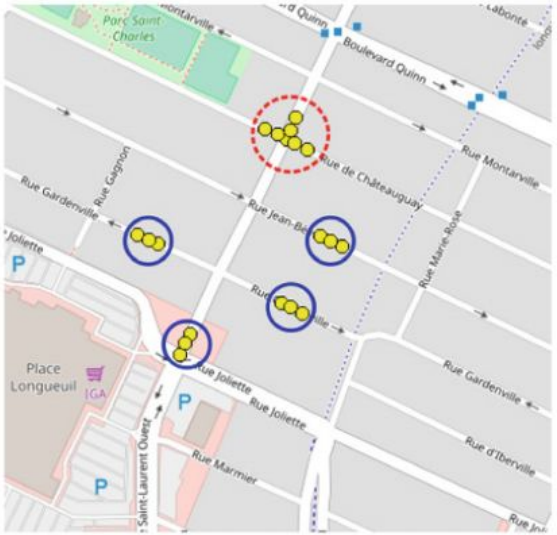
Algorithm 1. The particle filter algorithm

1: **function** PARTICLE FILTER($\mathcal{X}_{t-1}$, u_t, z_t)
2: $\quad \bar{\mathcal{X}} = \mathcal{X}_t = \emptyset$
3: $\quad$ **for** m = 1 to M **do**
4: $\quad\quad$ sample $x_t^{[m]} \sim p(x_t|u_t, x_{t-1}^{[m]})$
5: $\quad\quad$ $w_t^{[m]} = p(z_t|x_t^{[m]})$
6: $\quad\quad$ $\bar{\mathcal{X}}_t = \bar{\mathcal{X}}_t + \langle x_t^{[m]}, w_t^{[m]} \rangle$
7: $\quad$ **for** m = 1 to M **do**
8: $\quad\quad$ draw i with probability $\propto w_t^{[i]}$
9: $\quad\quad$ add $x_t^{[i]}$ to $\mathcal{X}_t$
10: $\quad$ **return** $\mathcal{X}_t$

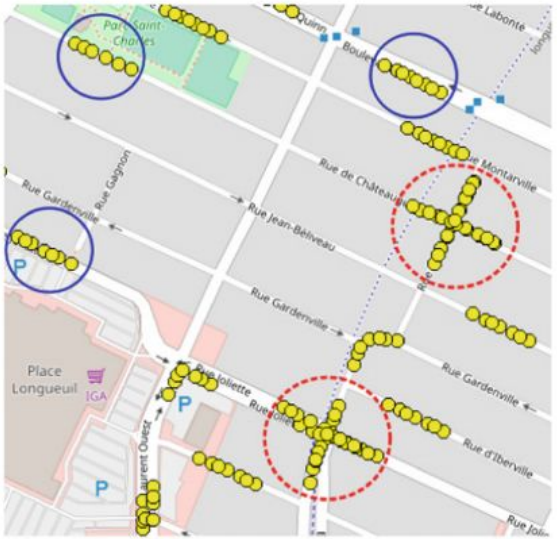
Model and Algorithm



(a) $t = 1$. The uncertainty on $s_1.p$ is low; few particles are needed.



(b) $t = 10$. $s_{10}.p$ can be 5 different positions.



(c) $t = 30$. More particles needed to represent the increased uncertainty

Model and Algorithm

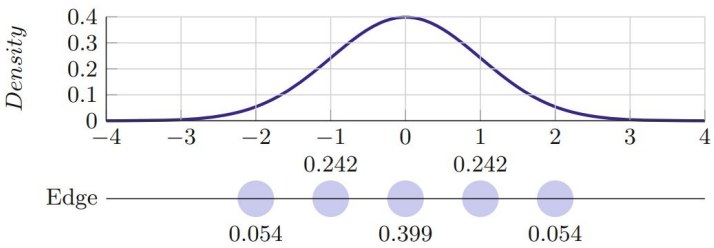
When to resample (i.e., do a new GPS reading)

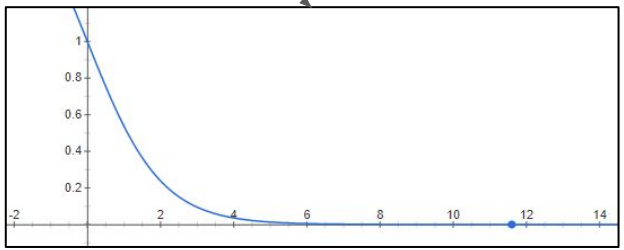
We define a *score(X_t)* function as the trade-off between the EC and the average estimated accuracy (TR and TMD)

$$\text{score}(\mathcal{X}_t) = \sum_{x \in \mathcal{X}_t} \Big(P(x) \times x_{acc} \Big) - \frac{2}{1 + e^{\lambda t}}.$$

Model and Algorithm

$$\text{score}(\mathcal{X}_t) = \sum_{x \in \mathcal{X}_t} \Big(P(x) \times x_{acc} \Big) - \frac{2}{1 + e^{\lambda t}}.$$



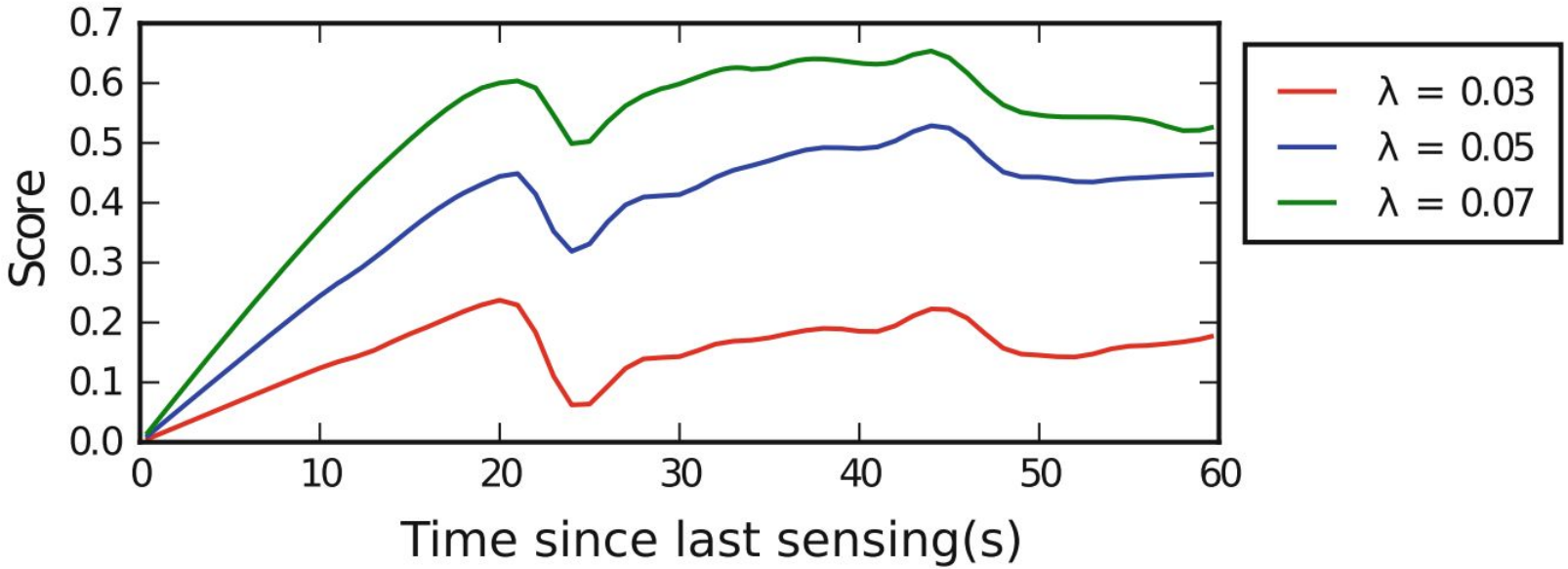


$$x_{acc} = \sum_{x' \in \bar{\mathcal{X}}_t} P(x') \times \frac{|x'_{Path^+}| + |x'_{Path^-}|}{|x_{Path}|},$$

- λ allows us to control how much importance to give to EC. The higher, the less important EC is.

Model and Algorithm

$$\text{score}(\mathcal{X}_t) = \sum_{x \in \mathcal{X}_t} \Big(P(x) \times x_{acc} \Big) - \frac{2}{1 + e^{\lambda t}}.$$



Model and Algorithm

- Resampling is done at : $t^* = \text{argmax}_t score(X_t)$
- The particles in a radius of $2\sigma^2$ are simply kept instead of generating new particles (where σ^2 is the radius (in meters) of 68% confidence given by the GPS sensor).
- The weight of each particle is then updated according to their distance from the GPS sensing.

Change of transport :

- Special nodes : bus stops(e.g. walk→bus), subway stations (walk→subway).
- During resampling : walk→car (can virtually happen anywhere, but that would lead to a combinatorial explosion)

Experimentations setup

Model data :

- Graph made with OpenStreetMap data
- Transit networks and schedules from GTFS data (agencies :STM, RTL, AMT, STL)
- Car speed : Open data collected by the City of Montréal with bluetooth devices

Experimentations data :

- Daily trips recorded with an Android app on an Asus Zenphone 4 Max smartphone
- All trips were collected at different time of the day and have different path.

Transport mode	Runtime (hours)	Number of GPS readings
Walk	8.03	28514
Car	11.94	42768
Bus	5.13	17290
Subway	2.06	390
Total	27.16	88962

Results - Accuracy

- λ controls how EC is important.
- Higher λs imply lower ASR (Average Sampling Rate)
- With λ = 0.01, better average accuracy (closest average accuracy is 94.4% [15]), but with a lower sampling rate

Table 3. Algorithm accuracy with different λ

λ	ASR (s)	Walk	Car	Bus	Subway	Avg TMD	TR Error
0.01	20.9	0.996	0.988	0.936	0.932	0.963	0.023
0.03	32.5	0.920	0.982	0.871	0.902	0.919	0.028
0.05	48.6	0.908	0.969	0.806	0.890	0.893	0.031
0.07	59.2	0.876	0.934	0.772	0.885	0.867	0.039
0.09	66.0	0.859	0.932	0.758	0.876	0.856	0.041

Results - Energy consumption

To measure EC, we used Android's BatteryManager API.

Before each test :

- Battery was fully charged
- All other applications were closed
- Wifi was disabled and 4G enabled

Method	λ	ASR (s)	EC (mW)
Fixed GPS sampling rate	-	1	336.82
	-	20	272.49
	-	40	172.67
	-	60	146.74
	-	80	114.02
Dynamic GPS sampling rate	0.01	20.9	286.28
	0.03	32.5	210.50
	0.05	48.6	170.79
	0.07	59.2	151.26
	0.09	66.0	148.18

Results - Energy consumption

- Most other approaches use an ASR of 1s → EC of 336 mW. (accuracy ranging from 84.0% to 94.4%)
- The presented approach :
  - ASR : 20.9s → EC of 286 mW
  - Accuracy of 96.3%

This means a 15.0% EC reduction for a 3.3% higher accuracy and a 37.5% EC reduction for an equivalent accuracy to Related Work.

Method	λ	ASR (s)	EC (mW)
Fixed GPS sampling rate	-	1	336.82
	-	20	272.49
	-	40	172.67
	-	60	146.74
	-	80	114.02
Dynamic GPS sampling rate	0.01	20.9	286.28
	0.03	32.5	210.50
	0.05	48.6	170.79
	0.07	59.2	151.26
	0.09	66.0	148.18

Results - Energy consumption

On lower ASR :

- ASR : 66.0s → EC of 148 mW
- Accuracy of 85.6%

This is a 56.0% EC reduction for an equivalent accuracy compared to [9].

Method	λ	ASR (s)	EC (mW)
Fixed GPS sampling rate	-	1	336.82
	-	20	272.49
	-	40	172.67
	-	60	146.74
	-	80	114.02
Dynamic GPS sampling rate	0.01	20.9	286.28
	0.03	32.5	210.50
	0.05	48.6	170.79
	0.07	59.2	151.26
	0.09	66.0	148.18

Conclusion

Using a particle filter, we achieved a better accuracy for a lower energy consumption on the TMD and TR problems.

The accuracy could be improved with a better transportation model :

- Elevation data
- Better traffic data
- Considering stops and lights

References

- 2. Bolbol, A., Cheng, T., Tsapakis, I., Haworth, J.: Inferring hybrid transportation modes from sparse GPS data using a moving window SVM classification. Comput. Environ. Urban Syst. 36(6), 526–537 (2012). https://doi.org/10.1016/j. compenvurbsys.2012.06.001
- 3. Byon, Y.J., Liang, S.: Real-time transportation mode detection using smartphones and artificial neural networks: performance comparisons between smartphones and conventional global positioning system sensors. J. Intell. Transp. Syst. 18(3), 264– 272 (2014). https://doi.org/10.1080/15472450.2013.824762
- 5. Carroll, A., Heiser, G., et al.: An analysis of power consumption in a smartphone. In: USENIX Annual Technical Conference, vol. 14, pp. 21–21 (2010)
- 8. Chung, E.H., Shalaby, A.: Transportation planning and technology a trip reconstruction tool for GPS-based personal travel surveys. Transp. Plan. Technol. 28(5), 381–401 (2005)
- 9. Dabiri, S., Heaslip, K.: Inferring transportation modes from GPS trajectories using a convolutional neural network. Transp. Res. Part C Emerg. Technol. 86, 360–371 (2018). https://doi.org/10.1016/j.trc.2017.11.021
- 10. Fang, S., Zimmermann, R.: EnAcq: energy-efficient GPS trajectory data acquisition based on improved map matching. In: Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 221–230 (2011). https://doi.org/10.1145/2093973.2094004
- 11. Li, X., Yuan, F., Lindqvist, J.: Feasibility of duty cycling gps receiver for trajectorybased services. In: 13th IEEE Annual Consumer Communications and Networking Conference (CCNC) (2016). https://doi.org/10.7282/T3VM4F56
- 13. Stenneth, L., Wolfson, O., Yu, P.S., Xu, B.: Transportation mode detection using mobile phones and GIS information. In: Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, p. 54 (2011). https://doi.org/10.1145/2093973.2093982
- 15. Xiao, G., Juan, Z., Gao, J.: Travel mode detection based on neural networks and particle swarm optimization. Information 6(3), 522–535 (2015). https://doi.org/10.3390/info6030522